

openQA Tests - action #54260

[kernel][s390x] Occasional failures due timeout

2019-07-15 10:06 - pvorel

Status:	Rejected	Start date:	2019-07-15
Priority:	High	Due date:	
Assignee:	cfconrad	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	QE Kernel Done		
Difficulty:			
Description			
<p>s390x sometimes dies during a test and we have no idea what is happening (no matter https://bugzilla.suse.com/show_bug.cgi?id=1140948 has been fixed and it's probably not caused by https://bugzilla.suse.com/show_bug.cgi?id=1136513 either).</p> <p>It can be anything between openqa, ssh, kernel.</p> <p>Here it looks like a timeout, but why? https://openqa.suse.de/tests/3055032#step/sem_unlink_2-2/7</p> <p>I'm going to test it on VM just with LTP (that's what I've done and nothing found on previous builds) and with my openQA instance (haven't tried yet). For both I'm going to use s390zp13.suse.de.</p> <p>Problematic tests:</p> <ul style="list-style-type: none">ltp_syscalls (https://openqa.suse.de/tests/3054999#step/prot_hsymlinks/7)ltp_controllers (https://openqa.suse.de/tests/3055029#step/cgroup_fj_stress_cpu_2_9_each/1)ltp_openposix (https://openqa.suse.de/tests/3055032#step/sem_unlink_2-2/6)ltp_cve, kotd_ltp_cve (https://openqa.suse.de/tests/3168071#step/cve-2011-2183/7, https://openqa.suse.de/tests/3168079#step/cve-2011-2183/7)install_ltp (found as problematic in the past, when serial console enabled, now working when disabled serial console with SERIAL_CONSOLE=0)			
Related issues:			
Related to openQA Tests - action #54275: [kernel]kotd[ltp][s390x] test fails...		Closed	2019-07-15
Related to openQA Tests - action #55883: known_issue:"Can.t fcntl" [kernel] l...		Resolved	2019-08-23
Related to openQA Tests - action #56036: [kernel][ltp] Fix killall call in cg...		Resolved	2019-08-28

History

#1 - 2019-07-15 10:26 - pvorel

[cfconrad](#) I wonder if this could be fixed with one of your PR

<https://github.com/os-autoinst/os-autoinst/pull/1166>

<https://github.com/os-autoinst/os-autoinst/pull/1165>

We wanted to fix failing install_ltp, but this might be similar: failing basically any long term running test in openQA on s390x. ltp_openposix and ltp_syscalls are hit by it. WDYT?

@all: I'll also try to clone these failing tests (ltp_openposix and ltp_syscalls) on osd with disabled serial console (it will be run old slow way with that console which uses needles). I guess it will finish, which will increase the need to fix serial console implementation (cfconrad is working on it).

#2 - 2019-07-15 10:28 - pvorel

As all jobs going longer than 30 mins fails, it might be some system timeout (ssh perl bindings?).

Than <https://github.com/os-autoinst/os-autoinst/pull/1166> could help.

#3 - 2019-07-15 11:11 - metan

This btw seems to be the same:

https://openqa.suse.de/tests/3055029#step/cgroup_fj_stress_cpu_2_9_each/1

And both tests seems to fail in random tests in the run but pretty reproducibly:

https://openqa.suse.de/tests/3055029#next_previous
https://openqa.suse.de/tests/3055032#next_previous

#4 - 2019-07-15 11:40 - pvorel

- Description updated

#5 - 2019-07-15 11:42 - pvorel

Cloned problematic jobs with disabled serial console (SERIAL_CONSOLE=0).
Tested with os-autoinst 4.5.1562413838.c3d5e8ac (very recent code).
Let's see whether serial console is to be blamed:

- ltp_syscalls
<https://openqa.suse.de/tests/3063485>
some failures, but test finished in 2,5 hours
- ltp_openposix
<https://openqa.suse.de/tests/3063480>
ok (in 4 hours)
- ltp_cve
<https://openqa.suse.de/tests/3063495>
ok
- ltp_controllers
<https://openqa.suse.de/tests/3063543>
https://openqa.suse.de/tests/3063543#step/cgroup_fj_stress_cpu_2_9_one/4
fails the same way even with serial console off
This test took too long to complete! It was running for 904.601083010435 seconds.

I should run ltp_controllers on that machine for longer time. And run the test again and connect to VM with virsh console --force while it's running to see what's going on.

#6 - 2019-07-15 12:29 - pvorel

Cloned problematic jobs on my openqa instance on s390zp13.suse.de

- ltp_syscalls
<http://quasar.suse.cz/tests/3229>
fails the same way as on osd
http://quasar.suse.cz/tests/3229#step/rt_sigaction02/7
- ltp_openposix
<http://quasar.suse.cz/tests/3228>
ok
- ltp_controllers
<http://quasar.suse.cz/tests/3236>
Failed on timeout (12 hrs)
http://quasar.suse.cz/tests/3236#step/cgroup_fj_stress_cpuacct_3_3_each/5
This job fails after 30 mins on osd (on s390x <https://openqa.suse.de/tests/3063543>), on other archs completes after 1,25 - 1,5 hour
- ltp_cve
<http://quasar.suse.cz/tests/3230>
ok

#7 - 2019-07-16 10:08 - pvorel

In case of ltp_controllers running in openQA, when observed (still running) job <http://quasar.suse.cz/tests/3323> on my instance s390zp13.suse.de on cgroup_fj_stress_freezer_2_2_each machine got irresponsible for some time due this test (but I haven't noticed high load before), so virsh console in svirt backend [1] exited.

In original draft of this feature I had more attempts to connect to console, which I removed as I thought this functionality is not needed.
We could add functionality:

after virsh console exits, inspect virsh list output to see if this VM is running. If yes, try to reconnect with virsh console several times (with sleep), it can take a while.

Or, maybe the whole concept with controlling VM with virsh console is wrong and we should rewrite it to use virtio socket + logging feature from qemu backend, which richie wrote 2:

```
-device virtio-serial -chardev socket,path=virtio_console,server,nowait,id=virtio_console,logfile=virtio_console.log,logappend=on \
```

```
-device virtconsole,chardev=virtio_console,name=org.openqa.console.virtio_console -chardev
socket,path=qmp_socket,server,nowait,id=qmp_socket,logfile=qmp_socket.log,logappend=on
```

But the question why it is slow and freeze remains. I'd be really happy if somebody more experienced in controllers could have look at the tests themselves and find whether the test is too crazy or it's really problem with our s390x kernel.

[1] <https://github.com/os-autoinst/os-autoinst/blob/master/backend/svirt.pm#L354>

[2] <https://github.com/os-autoinst/os-autoinst/blob/master/backend/qemu.pm#L845>

#8 - 2019-07-16 10:08 - pvorel

- Status changed from New to In Progress

#9 - 2019-07-16 10:14 - pvorel

Just for a record what we use now:

- 2 qemu processes created by qemu backend on x86_64 (have "logfile=virtio_console.log,logappend=on" and "-chardev socket,path=virtioconsole")

```
_openqa+ 12389 129 9.8 6532376 3206188 ? SI 12:04 10:13 /usr/bin/qemu-system-x86_64 -only-migratable -chardev
ringbuf,id=serial0,logfile=serial0,logappend=on -serial chardev:serial0 -soundhw ac97 -global isa-fdc.driveA= -m 4096 -cpu qemu64 -netdev
user,id=qanet0 -device virtio-net,netdev=qanet0,mac=52:54:00:12:34:56 -boot order=c,menu=on,splash-time=5000 -device usb-ehci -device
usb-tablet -smp 4 -enable-kvm -no-shutdown -vnc :91,share=force-shared -device virtio-serial -chardev
socket,path=virtio_console,server,nowait,id=virtio_console,logfile=virtio_console.log,logappend=on -device
virtconsole,chardev=virtio_console,name=org.openqa.console.virtio_console -chardev
socket,path=qmp_socket,server,nowait,id=qmp_socket,logfile=qmp_socket.log,logappend=on -qmp chardev:qmp_socket -S -device
virtio-scsi-pci,id=scsi0 -blockdev driver=file,node-name=hd0-overlay0-file,filename=/var/lib/openqa/pool/1/raid/hd0-overlay0.cache.no-flush=on
-blockdev driver=qcow2,node-name=hd0-overlay0,file=hd0-overlay0-file,cache.no-flush=on -device
virtio-blk,id=hd0-device,drive=hd0-overlay0,bootindex=0,serial=hd0 -blockdev
driver=file,node-name=cd0-overlay0-file,filename=/var/lib/openqa/pool/1/raid/cd0-overlay0.cache.no-flush=on -blockdev
driver=qcow2,node-name=cd0-overlay0,file=cd0-overlay0-file,cache.no-flush=on -device scsi-cd,id=cd0-device,drive=cd0-overlay0,serial=cd0
_openqa+ 12442 131 8.2 5964332 2683716 ? SI 12:04 10:16 /usr/bin/qemu-system-x86_64 -vga cirrus -only-migratable -chardev
ringbuf,id=serial0,logfile=serial0,logappend=on -serial chardev:serial0 -soundhw ac97 -global isa-fdc.driveA= -m 4096 -cpu qemu64 -netdev
user,id=qanet0 -device virtio-net,netdev=qanet0,mac=52:54:00:12:34:56 -boot order=c,menu=on,splash-time=5000 -device usb-ehci -device
usb-tablet -smp 4 -enable-kvm -no-shutdown -vnc :94,share=force-shared -device virtio-serial -chardev
socket,path=virtio_console,server,nowait,id=virtio_console,logfile=virtio_console.log,logappend=on -device
virtconsole,chardev=virtio_console,name=org.openqa.console.virtio_console -chardev
socket,path=qmp_socket,server,nowait,id=qmp_socket,logfile=qmp_socket.log,logappend=on -qmp chardev:qmp_socket -S -device
virtio-scsi-pci,id=scsi0 -blockdev driver=file,node-name=hd0-overlay0-file,filename=/var/lib/openqa/pool/4/raid/hd0-overlay0.cache.no-flush=on
-blockdev driver=qcow2,node-name=hd0-overlay0,file=hd0-overlay0-file,cache.no-flush=on -device
virtio-blk,id=hd0-device,drive=hd0-overlay0,bootindex=0,serial=hd0 -blockdev
driver=file,node-name=cd0-overlay0-file,filename=/var/lib/openqa/pool/4/raid/cd0-overlay0.cache.no-flush=on -blockdev
driver=qcow2,node-name=cd0-overlay0,file=cd0-overlay0-file,cache.no-flush=on -device scsi-cd,id=cd0-device,drive=cd0-overlay0,serial=cd0
-blockdev driver=file,node-name=cd1-overlay0-file,filename=/var/lib/openqa/pool/4/raid/cd1-overlay0.cache.no-flush=on -blockdev
driver=qcow2,node-name=cd1-overlay0,file=cd1-overlay0-file,cache.no-flush=on -device scsi-cd,id=cd1-device,drive=cd1-overlay0,serial=cd1
-blockdev driver=file,node-name=cd2-overlay0-file,filename=/var/lib/openqa/pool/4/raid/cd2-overlay0.cache.no-flush=on -blockdev
driver=qcow2,node-name=cd2-overlay0,file=cd2-overlay0-file,cache.no-flush=on -device scsi-cd,id=cd2-device,drive=cd2-overlay0,serial=cd2
```

- qemu process created by svirt on s390x

```
qemu 130712 6.4 7.2 2248232 895292 ? SI 04:11 8:01 /usr/bin/qemu-system-s390x -name guest=openQA-SUT-2,debug-threads=on -S
-object secret,id=masterKey0,format=raw,file=/var/lib/libvirt/qemu/domain-10-openQA-SUT-2/master-key.aes -machine
s390-ccw-virtio-3.1,accel=kvm,usb=off,dump-guest-core=off -m 1024 -realtime mlock=off -smp 2,sockets=2,cores=1,threads=1 -uuid
bdfbd56a-ba2e-43c8-a87a-1f3ca5226e4a -display none -no-user-config -nodefaults -chardev socket,id=charmonitor,fd=24,server,nowait -mon
chardev=charmonitor,id=monitor,mode=control -rtc base=utc -no-shutdown -boot strict=on -device virtio-serial-ccw,id=virtio-serial0,devno=fe.0.0002
-drive file=/var/lib/libvirt/images/openQA-SUT-2a.img,format=qcow2,if=none,id=drive-virtio-disk0,cache=unsafe -device
virtio-blk-ccw,scsi=off,devno=fe.0.0000,drive=drive-virtio-disk0,id=virtio-disk0,bootindex=1,write-cache=on -netdev
tap,fd=26,id=hostnet0,vhost=on,vhostfd=27 -device virtio-net-ccw,netdev=hostnet0,id=net0,mac=52:54:00:54:7a:00,devno=fe.0.0001 -chardev
pty,id=charconsole0 -device sclpconsole,chardev=charconsole0,id=console0 -chardev pty,id=charconsole1 -device
virtconsole,chardev=charconsole1,id=console1 -device virtio-balloon-ccw,id=balloon0,devno=fe.0.0003 -sandbox
on,obsolete=deny,elevateprivileges=deny,spawn=deny,resourcecontrol=deny -msg timestamp=on
```

#10 - 2019-07-16 10:27 - pvorel

I guess there must be a bug either in svirt backend or in /usr/share/openqa/script/worker (both perl). virsh console disconnected, but consoles::serial_screen::read_until() has not returned yet and /usr/share/openqa/script/worker is still busy (looking with strace it looks it runs ssh connection via libssh), until final timeout.

#11 - 2019-07-16 10:41 - pvorel

Running controllers on VM created with the same params as openQA instance to see if it manages to finish it.

#12 - 2019-07-16 10:43 - pvorel

- Description updated

#13 - 2019-07-16 20:56 - cfconrad

I'm debugging `consoles::serial_screen::read_until()` and try to understand what is going on there. For now I don't see any mysterious things going on, beside missing data on the `SSH::Channel` socket. I enabled `$ssh->debug(1)` on the ssh connection to see if there are some hints, but nothing. This is the commit which adds the debug info, maybe help full to understand the following analysis <https://github.com/cfconrad/os-autoinst/commit/df0ebc2c686a89885b81402605fd5eaab8015f1d>

see <http://cfconrad-vm.qa.suse.de/tests/5010/file/autoinst-log.txt>

```
* line 190522: enter the read_until() method
* line 201436 is a printf() when select returns on timeout
=> nothing mysterious in ssh debug

* line 201497 shows the output of `ps -aux` in such case, using a new SSH connection (0x52987bc0)
* line 201644 shows that our `script -f /tmp/serial_terminal.txt.rk4G5_lw7giu964L ...` is still running
=> so far so bad

* line 201902 shows the output of `cat /tmp/serial_terminal.txt.rk4G5_lw7giu964L` using a new SSH connection (
0x5298c7d0)
* line 213384 shows that the complete output of our failed command reached script and was written to that log
file
=> So we just missing it in STDOUT of script which is forwarded via SSH.
    For me, this proves that the command `virsh console openQA-SUT-9 console1` actually works in this test. But
    this needs further investigation!
```

```
* line 224878 shows the 'carry_buffer' which is the last thing which was read by `consoles::serial_screen`
* line 224928 is the result of an `index()` to find the `carry_buffer` inside the output of `cut /tmp/serial_t
erminal.txt.rk.....`
```

=> **Conclusion**, with this info it should be possible to implement a recover method. So we reenable the console. What I do not like is, that I still don't have a glue why we don't get the output of script anymore.

TODO:

- Check what happens if we avoid using script
- Check using `$ssh->$channel->pty('dump');`
- Check using `$ssh->$channel->shell();`
- POC for recover console()

#14 - 2019-07-17 21:20 - cfconrad

1st Error - NET::SSH2::Channel->read usage

One of our problems seems to be, that we used `NET::SSH2`. We used `NET::SSH2::sock()` with `select` for data indication and then did a `sysread` on the `NET::SSH2::Channel` object.

=> Now it happens that `select()` never returned but when I call `NET::SSH2::Channel->read()` in such case, I got the missing data! This PR solve this problem: <https://github.com/os-autoinst/os-autoinst/pull/1176>

I had two successful runs with this approach <http://cfconrad-vm.qa.suse.de/tests/5023> and <http://cfconrad-vm.qa.suse.de/tests/5020>, but now the test fails again, but with an other error picture.

2nd Error picture

This needs more investigation. For now I have only one assumption which is, that the LTP test really doesn't quit within 15min. For instance, when I'm comparing http://cfconrad-vm.qa.suse.de/tests/5026#step/cgroup_fj_stress_devices_2_2_each/6 with http://cfconrad-vm.qa.suse.de/tests/5023#step/cgroup_fj_stress_devices_2_2_each/8 I see that I miss some kills. And this is what I do miss all the time.

My current Questions:

- Is there a potential risk in that LTP tests, that they run longer then 15min or get stuck?
- Can I send a `SIGTERM` or other `SIG` which I can give by keyboard to kill that process?

TODO:

- Try kill on timeout like: https://github.com/os-autoinst/os-autoinst-distri-opensuse/blob/master/tests/publiccloud/run_ltp.pm#L101 or <https://github.com/os-autoinst/os-autoinst-distri-opensuse/blob/master/lib/publiccloud/provider.pm#L285>

#15 - 2019-07-23 09:38 - pvorel

- Assignee changed from pvorel to cfconrad

Assigning to Clements, as he's actually working on this.

#16 - 2019-07-23 11:54 - pcervinka

- Related to action #54275: [kernel]kotd][ltp][s390x] test fails in install_ltp - failure after install_kotd, console issue added

#17 - 2019-08-01 08:09 - pvorel

- Description updated

Added also ltp_cve and kotd_ltp_cve (running the same as ltp_cve)

#18 - 2019-08-01 11:39 - cfconrad

During tests, I encountered a problem with virtio_terminal that it drop random data with large output.

This PR use PIPE instead of UNIX sockets. Which let us better control the buffers <https://github.com/os-autoinst/os-autoinst/pull/1182>

Added a virtio_console test to check this: <https://github.com/os-autoinst/os-autoinst-distri-opensuse/pull/8092>

#19 - 2019-08-28 08:31 - cfconrad

- Related to action #55883: known_issue:"Can.t fcntl" [kernel] Investigate qemu left over, when switching from unix-domain sockets to named-pipes used for virtio_console handling added

#21 - 2019-08-28 09:40 - cfconrad

- Related to action #56036: [kernel][ltp] Fix killall call in cgroup_fj_stress_cpu added

#22 - 2019-08-28 09:45 - cfconrad

- Status changed from In Progress to Rejected

Close this ticket and follow up on more concrete once:

<https://progress.opensuse.org/issues/56033>

<https://progress.opensuse.org/issues/56036>

<https://progress.opensuse.org/issues/55883>

#23 - 2019-08-30 13:26 - jlausuch

- Target version changed from 445 to 457

#24 - 2020-10-23 13:39 - pcervinka

- Target version changed from 457 to QE Kernel Done